



TD N°3

Exercice 1

Afin de mieux répondre aux besoins de ses clients, la Société Tunisienne de Transport désire écrire un programme qui simule le comportement des voyageurs aux portillons automatiques placés à la sortie Place de Barcelone de la gare centrale de Tunis.

Le programme devrait comptabiliser deux éléments :

1. Le nombre total de voyageurs qui empruntent cette sortie à chaque jour. Ceci est réalisé avec la variable `Total_voyageurs`.
2. Le nombre courant de voyageurs qui empruntent cette sortie. Ceci est réalisé avec la variable `Courant_voyageurs`. Cette dernière sera incrémentée dès qu'un voyageur s'engage dans l'un des escaliers des quais menant à la sortie Place de Barcelone et décrétementée dès que le voyageur a passé un portillon automatique et a quitté la gare. Pour sortir de la gare, un voyageur doit passer son ticket par un portillon libre et il doit être, à un instant donné, le seul à utiliser ce portillon. À la sortie Place de Barcelone, il y a 3 portillons permettant aux voyageurs de sortir dans la rue.

Le pseudo code suivant montre le comportement d'un voyageur :

Processus Voyageur

Début

```
si (direction prise par le voyageur == sortie Place de
Barcelone)
    alors
        Total_voyageurs = Total_voyageurs + 1
        Courant_voyageurs = Courant_voyageurs + 1
        Passer_Portillon()
        Courant_voyageurs = Courant_voyageurs - 1
```

Fin

Synchronisez les processus Voyageurs au moyen de sémaphores. Pour ce faire, complétez le pseudo code ci-haut. Indiquez aussi le rôle et la valeur initiale de chaque sémaphore ainsi que les valeurs initiales des variables `Total_voyageurs` et `Courant_voyageurs`.

Exercice 2 : le coiffeur

Une illustration classique du problème de la synchronisation est celui du salon de coiffure. Dans le salon de coiffure, il y a un coiffeur C, un fauteuil F dans lequel se met le client pour être coiffé et N sièges pour attendre.

- S'il n'a pas de clients, le coiffeur C somnole dans le fauteuil F.
- Quand un client arrive et que le coiffeur C dort, il le réveille, C se lève. Le client s'assied dans F et se fait coiffer.
- Si un client arrive pendant que le coiffeur travaille :
 - si un des N sièges est libre, il s'assied et attend,
 - sinon il ressort.

Il s'agit de synchroniser les activités du coiffeur et de ses clients avec des sémaphores. Les sémaphores utilisés sont initialisés ainsi :

Init (SCF, 0);

Init (SP, 0);

Init (SX, 1);

	Programme client :	Programme coiffeur :
Client() {	<pre> P(SX); if(Attend < N) { Attend = Attend + 1; V (SP); V (SX); P (SCF); SeFaireCoifferEtSortir(); } else { V(SX); Sortir(); } </pre>	<pre> Coiffeur(){ while (1){ P(SP); P(SX); Attend = Attend -1; V(SCF); V(SX); Coiffer(); } </pre>
}		

1. Détailler le fonctionnement du coiffeur et de ses clients tels qu'ils sont représentés par les deux fonctions Coiffeur et Client.
2. Quel est le rôle de chacun des sémaphores SCF, SP et SX ?

Exercice 3 :

On considère les trois processus suivants qui s'exécutent de manière concurrente sur une machine. La variable **a** est une variable partagée, et on a les initialisations suivantes :
 int a = 6; Init(S1, 1); Init(S2, 0);

Processus A

```

P(S1);
a = a + 7;
V(S2);
    
```

Processus B

```

a = a - 5;
    
```

Processus C

```

V(S1);
P(S2);
a = a * 3;
    
```

1. Quelles sont les valeurs finales possibles de la variable a ? Donner pour chaque valeur la (les) suite(s) d'instruction qui amènent à cette valeur.
2. Ajoutez dans les programmes des processus les sémaphores nécessaires (et ceux-là seulement), pour obtenir dans toute exécution a = 34. Donnez les initialisations des sémaphores ajoutés.

Exercice 4

Un stand de fabrication et vente de crêpes s'installe sur un marché breton. Il ne vend qu'un produit : la « crêpe beurre sucre ». Le stand est tenu par deux personnes, et organisé de la façon suivante : – Le crêpier produit des crêpes (une par une) à partir d'un stock illimité de pâte à crêpe, et les dépose sur un tas de crêpes. Pour assurer la fraîcheur du produit, lorsque le tas de crêpes atteint 50 crêpes, le crêpier arrête momentanément de travailler. – Le vendeur reçoit la commande d'un client (un nombre n de crêpes, compris entre 1 et 10), prend les crêpes sur la pile et les donne au client (qui les prend et s'en va), puis passe au client suivant (on ne s'occupe pas du paiement). Le crêpier et le vendeur travaillent indéfiniment. Lorsqu'ils n'ont rien à faire, ils attendent. Par contre, chaque client achète une seule fois un nombre aléatoire de crêpes (entre 1 et 10). Après avoir demandé ses crêpes, le client doit attendre et partir avec son paquet de crêpes ! (Dans le code, le paquet sera symbolisé par un entier, égal au nombre de crêpes du paquet.)

1. Quels sont les processus dans ce problème ?

2. Quelles sont les ressources critiques ?
3. Donner la liste des sémaphores à utiliser pour pouvoir modéliser ce problème, avec pour chaque sémaphore sa valeur initiale (justifiée) et son rôle dans la solution.
4. Donner une solution à ce problème en utilisant des sémaphores. Pour ce faire, détailler le code de chacun des processus et donner le programme principal de l'application (l'initialisation des sémaphores)